ANÁLISIS NUMÉRICO I/ANÁLISIS NUMÉRICO – 2019
Trabajo de Laboratorio N$^{\underline{o}}$ 2

1. Escribir una función en `Julia` que implemente el método de bisección para hallar una raíz de $f : \mathbb{R} \mapsto \mathbb{R}$ en el intervalo $[a, b]$. La función debe llamarse "`rbisec`", y tener como entrada los argumentos `(fun,I,err,mit)`, donde `fun` es una función que dado x retorna $f(x)$, `I`$= [a, b]$ es un intervalo en $\mathbb{R}$, `err` es la tolerancia deseada del error y `mit` es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k-ésima iteración si $|f(x_k)| <$`err` o si $k \geq$`mit`. Los argumentos de salida deben ser `[hx,hf]` donde `hx`$= [x_1, \ldots, x_N]$ es el histórico de puntos medios y `hf`$= [f(x_1), \ldots, f(x_N)]$ el histórico de los respectivos valores funcionales.

2. Utilizar la función `rbisec` para:

   (a) encontrar la menor solución positiva de la ecuación $2x = \tan(x)$ con un error menor a 10^{-5} en menos de 100 iteraciones. ¿Cuántas iteraciones son necesarias cuando comenzamos con el intervalo $[0.8\,,\,1.4]$? Usar la siguiente sintaxis:
   `Julia> [hx,hy]=rbisec(@fun_lab2ej2a,[0.8,1.4],1e-5,100)`

   (b) Encontrar una aproximación a $\sqrt{3}$ con un error menor a 10^{-5}. Para esto, considere la función $x \mapsto x^2 - 3$ (que debe llamarse `fun_lab2ej2b`).

   (c) Graficar conjuntamente f y los pares $(x_k, f(x_k))$ para las dos funciones anteriores y con al menos dos intervalos iniciales distintos para cada una.

3. Escribir una función en `Julia` que implemente el método de Newton para hallar una raíz de $f : \mathbb{R} \mapsto \mathbb{R}$ partiendo de un punto inicial x_0. La función debe llamarse "`rnewton`", y tener como entrada `(fun,x0,err,mit)` donde `fun` es una función que dado x retorna $f(x)$ y $f'(x)$, `x0` es un punto inicial en $\mathbb{R}$, `err` es la tolerancia deseada del error y `mit` es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k-ésima iteración si vale alguna de las siguientes:

$$\frac{|x_k - x_{k-1}|}{|x_k|} < \texttt{err}, \qquad |f(x_k)| < \texttt{err}, \qquad k \geq \texttt{mit}.$$

   La salida debe ser `[hx,hf]` donde `hx`$= [x_1, \ldots, x_N]$ es el histórico de puntos generados y `hf`$= [f(x_1), \ldots, f(x_N)]$ el histórico de los respectivos valores funcionales.

4. Escribir dos funciones, `lab2ej4a` y `lab2ej4b`, que ingresando $a > 0$ retorne una aproximación de $\sqrt[3]{a}$. La aproximación debe realizarse usando el método de Newton del ejercicio anterior para resolver $x^3 - a = 0$ con un error menor a 10^{-6} mediante:

   (a) el uso de la función $x \mapsto x^2 - a$ (`fun_lab2ej4a`) donde a es definida como una variable global (use el comando `global a`),

   (b) el uso de la función $(x, a) \mapsto x^2 - a$ (`fun_lab2ej4b`) e invocando `rnewton` para la función `@(x)fun_lab2ej4b(x,a)`.

5. Escribir una función en `Julia` que implemente el método de iteración de punto fijo para hallar un punto fijo de $\varphi : \mathbb{R} \mapsto \mathbb{R}$ partiendo de un punto inicial x_0. La función debe llamarse "`ripf`", y tener como entrada `(fun,x0,err,mit)` donde `fun` es una función que dado x retorna $\varphi(x)$, `x0` es un punto en $\mathbb{R}$, `err` es la tolerancia deseada del error y `mit` es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k-ésima iteración si $|x_k - x_{k-1}| <$ `err` o $k \geq$ `mit`. La salida debe ser `hx` donde `hx`$= [x_1, \ldots, x_N]$ es el histórico de puntos generados.

6. Se quiere usar la fórmula de iteración $x_{n+1} = 2^{x_n - 1}$ para resolver la ecuación $2x = 2^x$. Utilizar la función del ejercicio anterior para investigar si converge; y en caso afirmativo, estudiar hacia quévalores lo hace para distintas elecciones de x_0, tomando un número máximo de 100 iteraciones y un error menor a 10^{-5}. Usar la siguiente sintaxis:
`Julia> hx=ripf(@fun_lab2ej6,x0,1e-5,100)`

7. Se desea conocer la gráfica de una función u definida implícitamente: $u(x) = y$ donde y es solución de
$$y - e^{-(1-xy)^2} = 0.$$
Implementar tres versiones de esta función, hallando el valor de y con los métodos de los ejercicios de biseccion (`lab2ej7bisec`), Newton (`lab2ej7newton`) y punto fijo (`lab2ej7ipf`). Los valores iniciales y tolerancias usadas por los distintos métodos deben ser escogidos de manera que cualquier usuario pueda graficar u en el intervalo $[0, 1.5]$ sin inconvenientes.